JOINT SOURCE-CHANNEL DECODING WITH ITERATIVE ALGORITHMS

Norbert Görtz and Ulrich Heute
Institute for Circuits and Systems Theory
Faculty of Engineering, Christian-Albrechts-University of Kiel
Kaiserstr. 2, D-24143 Kiel / Germany
e-mail: ng@techfak.uni-kiel.de

ABSTRACT

Ideas from iterative decoding of concatenated channel codes are adopted for joint source-channel decoding. Extrinsic information from a soft-in/soft-out channel decoder is used as a-priori information for the new soft-in/soft-out source decoder. The source decoder processes soft channel outputs, extrinsic information from the channel decoder, and source-statistics and it computes extrinsic information for the data-bits that again is used as a-priori information for the channel decoder. In this novel iterative approach to joint source-channel decoding the redundancies within the data-bits of the source encoder are used as an additional source of information in contrast to iterative channel decoding, where further information on the data-bits is drawn from the redundancy-bits of some additional, independent channel-coding scheme.

1 INTRODUCTION

Consider the problem of encoding and transmitting a source-signal vector U_k (k is the vector time-index, $N_U \in \mathbb{N}$ is the vector dimension). In practice source-encoding is often carried out in two steps: First the vector U_k of signal samples is decomposed into some (say $M \in \mathbb{N}$) parameter-vectors $X_k^{(1)}, ..., X_k^{(M)}$. A parameter-vector could consist of the LPC-coefficients[1] but the mean power of U_k could also be a (scalar) parameter. In the second step the vectors $X_k^{(j)}$ are quantized by N_j bits, $j = 1, ..., M$, and M indices $I_k^{(j)}$ are issued, which have to be transmitted to the source decoder. This scenario is depicted in figure 1.

A channel code is applied to improve the performance of the transmission system in presence of noise on the channel. The channel encoder computes codewords V_k with N_V bits from the N-bit vector I_k ($N < N_V$ and $N = \sum_j N_j$) which is generated from the output-indices $I_k^{(1)}, ..., I_k^{(M)}$ of the source encoder by the multiplexer. Nearly uncorrelated bit-errors after channel decoding are achieved by interleaving prior to the computation of the codeword. For the interleaved bit-vector the notation $\pi(I_k)$ is used.

In practical systems source coding is never perfect due to complexity limitations, delay constraints, and/or fixed lengths of the quantizer indices. Therefore some time-based dependencies between adjacent indices $I_{k-1}^{(j)}$, $I_k^{(j)}$ remain. In the following they are modeled by first-order stationary Markov-processes, which are described by the index-transition probabilities $P(I_k^{(j)} = \lambda \mid I_{k-1}^{(j)} = \mu)$ with $\lambda, \mu = 0, 1, \ldots, 2^{N_j} - 1$ and $j = 1, ..., M$. The probabilities can be measured and stored in advance. In order to simplify the realization of iterative source-channel decoding the source-encoder indices are assumed to be *mutually* independent, which at least is approximately fulfilled in many practical situations.

In the model in figure 1 the codeword is transmitted over a discrete AWGN-channel. Coherently detected binary phase-shift keying is assumed for the modulation. The transmission of a single bit is modeled by the following procedure: The bits $v_{l,k} \in \{0, 1\}$, $l = 1, 2, ...N_V$, are first mapped to "bipolar" bits $v'_{l,k} \in \{+1, -1\}$, i.e. $v'_{l,k} = 1 - 2 \cdot v_{l,k}$. After that a white Gaussian noise-sample $n_{l,k}$ is added to $v'_{l,k}$. The noise has zero mean and a variance of $\sigma_n^2 = \frac{N_0}{2E_s}$, where E_s is the energy that is used to transmit each bit (data bits and redundancy bits) and N_0 is twice the noise power spectral density $N_0/2$ on the (white noise) channel. The conditioned p.d.f. of the received value $\tilde{v}_{l,k}$ at the channel output, given that the bit $v_{l,k} \in \{0, 1\}$ has been transmitted, is given by

$$p(\tilde{v}_{l,k}|v_{l,k}) = \frac{e^{-\frac{1}{2\sigma_n^2}(\tilde{v}_{l,k} - v'_{l,k})^2}}{\sqrt{2\pi}\sigma_n} \Big|_{v'_{l,k} = 1 - 2 \cdot v_{l,k}} . \quad (1)$$

2 ITERATIVE SOURCE-CHANNEL DECODING

The basic idea of iterative source-channel decoding is adopted from iterative decoding of concatenated channel codes: The redundancies, which are contained in the channel code-words and in the source-encoder indices, are exploited alternately by separate soft-in/soft-out decoders (SISO decoders) for the channel-code and the source-code. Each SISO decoder computes the new (extrinsic) part of information on the data-bits, which is based only on *one* type of redundancy. The extrinsic information is forwarded to the other SISO decoder as a-priori information. This process is iteratively repeated to improve the reliability of the decoded index-bits step by step. A block-diagram of such an iterative source-channel decoder is depicted in figure 2. It directly fits into the model of the transmission system in figure 1.

2.1 Soft-In/Soft-Out Channel Decoder

SISO channel decoders have been treated in detail e.g. in [1], so here only the interface to the new SISO source decoder is described.

In figure 2 information is passed between the blocks by log-likelihood ratios (L-values, "soft-bits") [1]. The a-posteriori log-likelihood ratio of an (bipolar) output-bit $i'_{l,k}$ of the SISO

[1] The LPC-coefficients (**L**inear **P**redictive **C**oding) describe the spectral shape of a source signal.

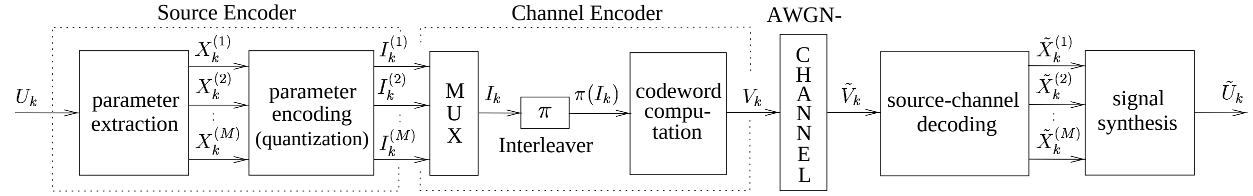

Figure 1: Model of a transmission system with M source indices

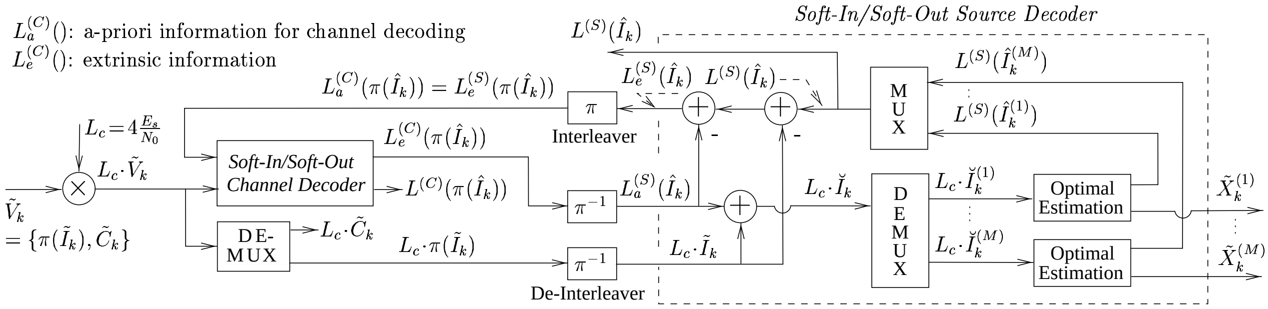

Figure 2: Iterative source-channel decoding with soft-in/soft-out decoders

channel decoder is defined by

$$L^{(C)}(\hat{i}_{l,k}) \doteq L(i'_{l,k}|\tilde{V}_k) = \log \frac{P(i'_{l,k}=+1|\tilde{V}_k)}{P(i'_{l,k}=-1|\tilde{V}_k)} \qquad (2)$$

where "log" denotes the natural logarithm and $l = 1, 2, ..., N$. The L-value $L^{(C)}(\hat{i}_{l,k})$ is "conditioned" on the received channel word $\tilde{V}_k$ at the current time k, so the whole redundancy of the channel-codewords is exploited by the SISO channel decoder to estimate each output bit $i'_{l,k}$. The a-priori-information $L_a^{(C)}(\hat{i}_{l,k})$ of the index-bit is included in (2). The notation $L^{(C)}(\hat{I}_k)$ is an abbreviation for a vector of L-values which is given by

$$L^{(C)}(\hat{I}_k) = \left\{L^{(C)}(\hat{i}_{1,k}), \ldots, L^{(C)}(\hat{i}_{N,k})\right\}. \qquad (3)$$

The interleaver $\pi()$ only changes the order of the N L-values in (3). Therefore it was left away in the notation above. The L-values with low reliabilities (small absolute values), which appear in bursts at the output of the SISO *channel* decoder for certain channel codes, are spread by the interleaver, i.e. after de-interleaving the L-values seem to be independent for the SISO *source* decoder. This corresponds to the assumption of a white-noise channel in the derivation of Optimal Estimation [2], which will be used for SISO *source* decoding.

The SISO channel decoder accepts a-priori information $L_a^{(C)}(\pi(\hat{I}_k))$ of the index-bits, which is initially set to zero in the first iteration, and the received channel-L-values $L_c \cdot \tilde{V}_k$ of the transmitted codeword-bits and it computes the output L-values $L^{(C)}(\pi(\hat{I}_k))$ of the index-bits. The channel-reliability is given by $L_c = 4 \cdot E_s/N_0$ $\big(E_s/N_0$ must be known, which will also be required for SISO source decoding$\big)$. The extrinsic information $L_e^{(C)}(\pi(\hat{I}_k))$ contains the *new* part of information which has been determined by the SISO channel decoder by only exploiting the redundancies within the channel codewords. The extrinsic information can be computed by

$$L_e^{(C)}(\pi(\hat{I}_k)) = L^{(C)}(\pi(\hat{I}_k)) - L_c \cdot \pi(\tilde{I}_k) - L_a^{(C)}(\pi(\hat{I}_k)). \quad (4)$$

This formula implicates that a systematic channel code is used, since we assume that we know the received L-values $L_c \cdot \pi(\tilde{I}_k)$ of the index-bits. These L-values are also required for the SISO source decoder, i.e. in this paper iterative source-channel decoding is stated for *systematic* channel codes.

2.2 Soft-In/Soft-Out Source Decoder

SISO source decoding is performed by the Optimal-Estimation-algorithm (OE) described in [3], [2]. It processes the channel-values $\tilde{I}_k$ of the index-bits and the a-priori information $L_a^{(S)}(\hat{I}_k)$, which equals the de-interleaved extrinsic information $L_e^{(C)}(\hat{I}_k)$ from the channel-decoder. OE exploits these informations and the correlation of the source-encoder indices in order to compute the a-posteriori probabilities (APPs) of the parameter indices, which are defined by:

$$P_{AP}^{(j)}(\lambda, k) \doteq P\Big(I_k^{(j)} = \lambda \mid \check{I}_k^{(j)}, \check{I}_{k-1}^{(j)}, \ldots\Big), \qquad (5)$$

for $\lambda = 0, 1, \ldots, 2^{N_j} - 1$ and $j = 1, ..., M$. The vector $\check{I}_k$ of improved channel-values for the data bits is computed by

$$\check{I}_k \doteq \tilde{I}_k + L_a^{(S)}(\hat{I}_k)/L_c\ ,\ L_c = 4 \cdot E_s/N_0\ , \qquad (6)$$

using the a-priori information $L_a^{(S)}(\hat{I}_k)$ for the current iteration from the channel decoder, i.e. the channel-values (6) are different in each iteration and they should become more reliable with an increasing number of iterations. The associated values $\check{I}_k^{(j)}$ for the indices are obtained after demultiplexing (see figure 2). In [2] it is shown that the APPs can be computed by the following recursion:

$$P_{AP}^{(j)}(\lambda, k) = M_k^{(j)} \cdot \overbrace{p\big(\check{I}_k^{(j)} \mid I_k^{(j)} = \lambda)\big)}^{\text{product of (1) over } l=1,\ldots,N_j \text{ with } \check{v}_{l,k}=\check{i}_{l,k}} \cdot \sum_{\mu=0}^{2^{N_j}-1} \underbrace{P\big(I_k^{(j)} = \lambda \mid I_{k-1}^{(j)} = \mu\big)}_{\text{trans. probabilities, Markov-source}} \cdot P_{AP}^{(j)}(\mu, k-1) \qquad (7)$$

with $\lambda = 0, 1, \dots, 2^{N_j} - 1$ and $j = 1, ..., M$. The factor $M_k^{(j)}$ normalizes the sum in (7) in such a way, that $P_{AP}^{(j)}(\lambda, k)$ is a true probability that sums up to 1 over all indices λ. The APPs (5) are computed independently for each of the M parameter-indices. The APPs are used to estimate the output-parameters $\tilde{X}_k^{(j)}$ only after the last iteration. If the mean squared error of the *parameters* is an appropriate criterion for the "overall" quality of the transmission (this is assumed in the following) mean-square estimators are used for the computation of the parameters in the decoder, i.e.:

$$\tilde{X}_k^{(j)} = \sum_{\lambda=0}^{2^{N_j}-1} X_q^{(j)}(\lambda) \cdot P_{AP}^{(j)}(\lambda, k) \qquad (8)$$

where $X_q^{(j)}(\lambda)$ is the entry with the number λ of the quantization table for the parameter-vector $X_k^{(j)}$. The estimated parameters are used to synthesize the actual decoder-output signal U_k. In general this procedure will not lead to the optimum SNR with respect to the decoder-output signal U_k, but in many practical cases a close approximation of the *parameters* (e.g. in the mean-square-sense) will also lead to a good overall-quality of the transmission.

Within the iterations the extrinsic information, which is used as a-priori information for the channel decoder, must be computed from the *temporary* values of the APPs. The SISO channel decoders, which are available for binary channel-codes, require a-priori information for *single data-bits*, i.e. we would like to have

$$L^{(S)}(\hat{i}_{l,k}^{(j)}) = \log \frac{P\big(i_{l,k}^{'(j)} = +1 \mid \check{I}_k^{(j)}, \check{I}_{k-1}^{(j)}, \dots\big)}{P\big(i_{l,k}^{'(j)} = -1 \mid \check{I}_k^{(j)}, \check{I}_{k-1}^{(j)}, \dots\big)} \qquad (9)$$

available. The quantity (9) is the log-likelihood ratio of the output-bit $i_{l,k}^{'(j)}$ of the SISO source decoder. However, optimal estimation computes APPs of indices, i.e. information on *groups* of bits. So we have to convert index-probabilities into bit-probabilities. The solution to this problem is given by

$$L^{(S)}(\hat{i}_{l,k}^{(j)}) = \log \frac{P_1^{(j)}(\lambda_l = 0)}{P_1^{(j)}(\lambda_l = 1)} \qquad (10)$$

with

$$P_1^{(j)}(\lambda_l = \xi) = \sum_{\substack{\lambda=\{\lambda_1,\dots,\lambda_{N_j}\},\\ \lambda\in\{0,1\}^{N_j} \wedge \lambda_l=\xi}} P_{AP}^{(j)}(\lambda, k)\ , \quad \xi \in \{0, 1\}\ , \qquad (11)$$

for all the bits indexed by $l = 1, ..., N_j$, where the indices λ consist of binary bits $\lambda_{l'} \in \{0, 1\}$, $l' = 1, 2, \dots, N_j$, i.e. in (10) the bipolar bits were mapped to "normal" binary bits. It should be noted that the summation over all probabilities corresponding to the indices with a "1" or "0" at bit-position l "throws away" the correlation of the index-bits, i.e. a lot of information gets lost due to the fact that we are separately looking at the probabilities of single bits without considering their dependencies on other bits. As a consequence not all the "new" information is forwarded to the channel decoder and this is the reason why iterations can improve the performance. If we had assumed that all indices have the same number of bits, i.e. $N_j = N_1, \forall j$, then the symbol-alphabet of a (non-binary) channel code could be adapted to the number N_1. In this case no iterations would be necessary since all the information due to the index-correlation could be forwarded to the channel decoder in a single step, which would result in a non-binary version of source-controlled channel decoding [4]. One problem of this approach is the tremendous complexity which is required for the decoding of non-binary channel-codes with code-symbols consisting of 8 bits for instance. The latter is a usual number of index-bits e.g. in speech coding. Another problem is due to the fact that only one type of code-symbols can be used, i.e. all indices would be forced to have the same number of bits. The advantage of the method stated in this paper is that the correlations of source-encoder indices with variable and possibly moderate-to-large numbers of bits (as in speech and audio coding) can be exploited for channel decoding with realizable complexity.

After computing the L-values of the index-bits by (10) and (11), the output-L-values $L^{(S)}(\hat{I}_k)$ of the SISO source decoder are obtained by appropriate multiplexing. The extrinsic information is computed by

$$L_e^{(S)}(\hat{I}_k) = L^{(S)}(\hat{I}_k) - L_c \cdot \tilde{I}_k - L_a^{(S)}(\hat{I}_k) \quad . \qquad (12)$$

It is interleaved and forwarded to the SISO channel decoder as the a-priori information $L_a^{(C)}(\tilde{\pi}(I_k))$.

3 SIMULATION RESULTS

The source-encoder was simulated by $M = 9$ parameters $X_k^{(j)}$, $j = 1, 2, ..., M$, which are assumed to appear in parallel at the output of the source-encoder at each time k. The performances of the source-channel decoding systems were evaluated by their mean *parameter*-SNRs.

Gaussian random signals were correlated by low-pass filters with the transfer functions $H(z) = \frac{(1-a)\cdot z}{z-a}$, $a = 0.9$, to generate model-parameter-signals $X_k^{(j)}$. As parameter encoders 5-bit optimal scalar quantizers (i.e. $N_j = 5\ \forall j$) were used, which issue the indices $I_k^{(j)}$. The total number of index-bits was $N = 9 \cdot 5 = 45$.

The source-encoder output-bits $I_k = \{I_k^{(1)}, ..., I_k^{(M)}\}$ were spread by a new random-interleaver at each time k and afterwards they were commonly channel-encoded by a memory-2 rate-1/2 recursive systematic convolutional code (RSC-code, [5]). The channel code was terminated by two tail-bits, so the total number of bits in the codewords was $2 \cdot (N + N_{Tail}) = 2 \cdot (45 + 2) = 94$ resulting in an overall code-rate of $R = 45/94 \approx 0.5$.

The codewords were transmitted over the AWGN-channel and the new iterative decoding-technique was applied. For the SISO channel decoding of the RSC-code the BCJR-algorithm [1], [6] was used and Optimal Estimators (OE) were applied as SISO source decoders for the parameters. In the following the iterative source-channel decoding-technique is denoted by "BCJR-OEms, ρ it.", where $\rho = 1, 2, 4$ is the number of iterations. One iteration consists of SISO channel decoding (with zero a-priori information $L_a^{(C)}(.)$ in the first iteration) followed by optimal estimations, in which the extrinsic information from channel decoding is exploited. As described above, extrinsic information is computed by the OE-algorithms which is used as a-priori information for the channel-decoder in the next iteration. Actually "BCJR-OEms, **1 it.**" is no iterative decoding but "standard" sequential decoding for channel and source.

The results of the simulations are depicted in figure 3. The mean parameter-SNR (averaged over $M = 9$ indices) is plotted over E_b/N_0, the ratio of the energy E_b per transmitted *data*-bit and twice the noise power spectral density $N_0/2$ on the AWGN-channel. Mostly MS-estimators (denoted by "ms") were used to compute the parameter-estimates $\tilde{X}_k^{(j)}$ from the APPs after the last iteration.

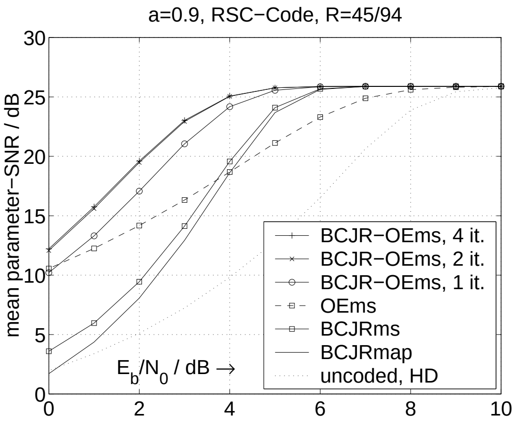

Figure 3: Performance of iterative source-channel decoding. Transmission of $M=9$ *strongly* autocorrelated parameters ($a=0.9$) in parallel. Parameter encoding by optimal scalar 5-bit quantizers.

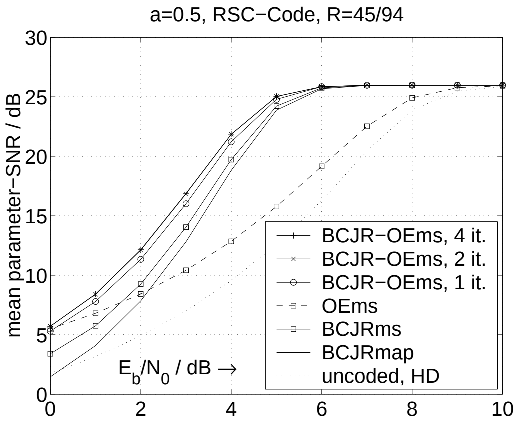

Figure 4: Performance of iterative source-channel decoding. Transmission of $M=9$ *moderately* autocorrelated parameters ($a=0.5$) in parallel. Parameter encoding by optimal scalar 5-bit quantizers.

The performance of the new iterative procedure was also compared with channel decoding by the BCJR-algorithm with a) hard decisions afterwards, which is denoted by "BCJRmap", and b) with using the soft-outputs of the BCJR-algorithm directly in MS-estimators for the output-parameters $\tilde{X}_k^{(j)}$ (curve labeled "BCJRms") without exploiting the source-correlation and/or the probability distribution of the source-encoder indices.

Additionally the performance of Optimal Estimation with a MS-estimator (curve labeled "OEms") without exploiting the channel code was simulated. The performance of a system using hard-decisions for the data-bits directly at the channel output is given by the curve labeled "uncoded, HD".

Figure 3 shows that "BCJR-OEms" works better than all other algorithms. Using two iterations instead of one improves the mean parameter-SNR by about 2.5 dB for $E_b/N_0 < 3$ dB which is equivalent to a possible reduction of the transmission power by 0.8 dB. More than two iterations do not further improve the performance significantly.

The BCJRms-algorithm works better than OEms down to $E_b/N_0 \approx 4$ dB. For channels worse than this, i.e. $E_b/N_0 <$ 4 dB, the performance of "BCJRms" strongly degrades. In this area "OEms" works better because the strong source-correlation allows a good prediction of the parameters in case of unreliable channel-values for the index-bits. The iterative BCJR-OEms-algorithm exploits both sources of information so the performance for moderately corrupted channels (1 dB $< E_b/N_0 <$ 5 dB) is significantly better than for the separately used component-algorithms "BCJR" and "OEms" without iterations.

If the source-signals are weaker correlated, the results, which are depicted for $a = 0.5$ in figure 4, are qualitatively similar but the correlation does not allow a signal prediction as good as in the previous case. Therefore the performances of all schemes that exploit the source correlation are worse, but still "BCJR-OEms" works best of all.

4 CONCLUSIONS

A new iterative source-channel decoding-technique was stated which uses a soft-in/soft-out *channel* decoder and a soft-in/soft-out *source* decoder. Both SISO decoders operate alternately and they interchange extrinsic information on the data-bits which is drawn either from the redundancies within the channel-codewords or from the correlation of the source-encoder indices.

SISO channel decoders are known from literature while the SISO source decoders, which are based on optimal parameter-estimations, and especially their interfaces to SISO channel decoders are newly described in this paper. Due to the iterative processing of source- and channel-code-redundancies the new algorithm can use powerful binary channel codes, which commonly encode all the parameter-indices of a source-encoder. The indices are allowed to consist of different and possibly large numbers of bits.

The performance of the new iterative source-channel decoding is significantly better than that of the sequentially applied component algorithms without any iterations.

REFERENCES

[1] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 429–445, Mar. 1996.

[2] N. Görtz, "Joint source-channel decoding by channel-coded optimal estimation," in *Proceedings of the 3rd ITG Conference Source and Channel Coding*, Jan. 2000, pp. 267–272.

[3] T. Fingscheidt and P. Vary, "Robust speech decoding: A universal approach to bit error concealment," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Apr. 1997, vol. 3, pp. 1667–1670.

[4] J. Hagenauer, "Source-controlled channel decoding," *IEEE Transactions on Communications*, vol. 43, no. 9, pp. 2449–2457, Sept. 1995.

[5] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Transactions on Communications*, vol. 44, no. 10, pp. 1261–1271, Oct. 1996.

[6] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Transactions on Information Theory*, vol. IT-20, pp. 284–287, Mar. 1974.